

BOOSTING-BASED DDOS DETECTION IN INTERNET OF THINGS SYSTEMS

Er.R.Abhinaya,

Dept. of Computer Science

Sri Vidya College of Engineering & Technology,

Sivakasi, Tamil Nadu - 626005.

abinayaramesh332001@gmail.com

Mrs. M. Mohana M.E.,

Assistant Professor : dept .of Computer Science,

Sri Vidya College of Engineering & Technology,

Sivakasi, Tamil Nadu -626005.

m.mohanamo@gmail.com

Abstract-

Distributed denial of service (DDoS) attacks remain challenging to mitigate in existing systems, including in-home networks that comprise different Internet of Things (IoT) devices. In this paper, we present a DDoS traffic detection model that uses a boosting method. Devices have also been exploited to create a bot net network to generate distributed denial of service (DDoS) traffic. There have been many applications of machine learning techniques to detect DDoS traffic, which can be categorized into those based on supervised techniques (using existing knowledge to classify future unknown instances) and those based on unsupervised techniques (trying to determine the corresponding instance class without prior knowledge). Even though advanced Machine Learning (ML) and deep learning techniques have been adopted for DDoS detection, the attack remains a major threat of the Internet. The boosting learning classification algorithm is used for classifying the data that is presented in the network. Existing public datasets were used to evaluate the detection model. The Main aim of this project is identifying or detecting the attacks which in occurred in the network by using the various classification algorithms. Now growth of social network will get increased in every day to day basis. However, it is a challenging issue to detect the attacks. In our process, the system is developed five different machine and deep learning algorithms for detecting the ddos attack.

1.INTRODUCTION

Internet of Things (IoT) devices and systems are becoming commonplace, and hence they are increasingly targeted by attackers, for example, by identifying and exploiting vulnerabilities in IoT software and hardware, or their implementation, to facilitate unauthorized and malicious activities. Such devices have also been exploited to create a botnet network to generate distributed denial of service (DDoS) traffic. DDoS represents a critical network oriented cyber threat, whose trend has been steadily rising over the last decade. For example, the DDoS attacks targeting Amazon AWS in Q1 of 2020 reportedly had a peak volume of 2.3 Tbps. IoT devices and systems are found not only in an organizational or government setting, but also in our homes.

Smart homes are one of the fastest-growing IoT applications, and the deployed devices are extremely heterogeneous. Such devices are often shipped with minimal or non-existent security mechanisms, and in an effort to make these devices user friendly the security requirements are often reduced. In addition, many of the devices in a smart home are inexpensive and do not have significant computational capabilities, and consequently, they can be easily compromised to facilitate a broad range of nefarious activities, including generating DDoS traffic. In a typical smart home ecosystem, there are several stakeholder groups such as end-users (homeowners or tenants within a home), Internet/telecommunication service providers, device manufacturers, and service providers (e.g., third-party service providers such as a monitored security service). These stakeholders generally have a vested interest not to be involved in malicious cyber activities, or for their devices, systems, platforms, and/or infrastructure to be exploited to facilitate nefarious activities. For example, it is in the interest of Internet/telecommunication service providers to promptly detect any unauthorized behavior/activities within a smart home environment, to protect their own network infrastructure and prevent the compromised devices / systems to be used as a launch pad against other devices and systems (with associated legal and financial implications). For the effective feature selection and accurate Bot-IoT attacks identification in IoT network environment a new develop dataset is used. The dataset includes on the Internet of Things, and normal traffic flows as well as several numerous cyber-attacks traffic flows of botnets attacks. To trace the accurate traffic and develop effective dataset, the realistic test bed is used for the development of this dataset with effective information features. Similarly, for the improvement of machine learning model performance and effective prediction model, more features were extracted and added with extracted features set. However, for better performance results, the extracted features are labelled, such as attack flow, categories, and subcategories. Nowadays, the Internet of Things (IoT) technology is growing up more day by day, and in every minute, numerous devices are getting connected with this technology. By using this technology,

daily life becomes more convenient and well-organized. For instance, initially, IoT technology was limited to small offices and homes, but nowadays, IoT technology integrated into industries for more reliability and saving time. However, IoT technology is becoming an essential part of our daily life. In 2021, the IoT technology will grow up, and more than 27 million IoT devices will connect, which will be a tremendous change in IoT technology world. With the rapid development and popularization of Internet of Things (IoT) devices, an increasing number of cyber-attacks are targeting such devices. It was said that most of the attacks in IoT environments are botnet-based attacks. Many security weaknesses still exist on the IoT devices because most of them have not enough memory and computational resource for robust security mechanisms. Moreover, many existing rule-based detection systems can be circumvented by attackers. In this study, we proposed a machine learning (ML)-based botnet attack detection framework with sequential detection architecture. An efficient feature selection approach is adopted to implement a lightweight detection system with a high performance. Botnet is a network of numerous bots designed to perform malicious activities on the target network which are controlled using command and control protocol by the single unit called botmaster. Bots are the infected computers controlled remotely by the botmaster without any sign of being hacked and are used to perform malicious activities. Botnet size varies from small botnet consists of few hundred bots to the large botnets with 50,000 hosts. Hackers spread botnet malware and operate secretly without any noticeable indication of their presence and can remain effective and functioning for years. To secure connected IoT devices against complex botnet attacks, Machine Learning (ML) techniques have been employed to develop Network Intrusion Detection Systems (NIDS). Such NIDS can be installed at strategic points within an IoT network. Specifically, Deep Learning (DL), an advanced ML approach, offers a unique capability for automatic extraction of features from large-scale, high-speed network traffic generated by interconnected heterogeneous IoT devices. Considering the resource-constraints in IoT devices, NIDS techniques used in classical computer networks are not efficient for botnet detection in IoT systems due to high computation and memory requirements. In order to develop an efficient DL method for botnet detection in IoT networks, sufficiently large network traffic information is needed to guarantee efficient classification performance. However, processing and analyzing high-dimensional network traffic data can lead to curse of dimensionality. Also, training DL models with such high-dimensional data can cause Hughes phenomena. High-dimensional data processing is complex and requires huge computational resources and storage capacity. IoT devices do not have sufficient memory space to store big network traffic data required for DL. Therefore, there is a need for end-to-end DL-based botnet detection method that will reduce high dimensionality of big network traffic features and also detect complex and recent botnet attacks accurately based on low-dimensional network traffic information. Currently, Bot-IoT dataset is the most relevant publicly available dataset for botnet attack detection in IoT networks because it: (a) has IoT network traffic samples; (b) captured complete network

information; (c) has a diversity of complex IoT botnet attack scenarios; (d) contains accurate ground truth labels; and (e) provides massive volume of labeled data required for effective supervised DL. The original feature dimensionality¹ of the Bot-IoT dataset is 43, and the memory space required to store this network traffic data is 1.085 GB. So far, feature dimensionality reduction methods that have been applied to the Bot-IoT dataset were all based on feature selection techniques.

2.LITERATURE SURVEY

1.DDoS Detection System: Using a Set of Classification Algorithms Controlled by Fuzzy Logic System in Apache Spark, 2019

Author: Amjad Alsirhani, Srinivas Sampalli, Peter Bodorik

Methodology:

Distributed denial of service (DDoS) attacks are a major security threat against the availability of conventional or cloud computing resources. Numerous DDoS attacks, which have been launched against various organizations in the last decade, have had a direct impact on both vendors and users. Many researchers have attempted to tackle the security threat of DDoS attacks by combining classification algorithms with distributed computing. However, their solutions are static in terms of the classification algorithms used. In fact, current DDoS attacks have become so dynamic and sophisticated that they are able to pass the detection system thereby making it difficult for static solutions to detect. In this paper, we propose a dynamic DDoS attack detection system based on three main components: 1) classification algorithms; 2) a distributed system; and 3) a fuzzy logic system. Our framework uses fuzzy logic to dynamically select an algorithm from a set of prepared classification algorithms that detect different DDoS patterns. Out of the many candidate classification algorithms, we use Naive Bayes, Decision Tree (Entropy), Decision Tree (Gini), and Random Forest as candidate algorithms. We have evaluated the performance of classification algorithms and their delays and validated the fuzzy logic system.

2.Measuring the Effects of Data Parallelism on Neural Network Training,2018

Author: Christopher J. Shallue, Jaehoon Lee, Joseph Antognini, JaschaSohl-Dickstein, Roy Frostig, George E. Dahl.

Methodology:

Our experimental data is publicly available as a database of 71,638,836 loss measurements taken over the course of training for 168,160 individual models across 35 workloads. Data communication networks typically consist of end-user devices, or hosts interconnected by the network infrastructure. This infrastructure is shared by hosts and employs switching elements such as routers and switches as well as communication links to carry data between hosts. Routers and switches are usually “closed” systems, often with limited-and vendor-specific control interfaces. Therefore, once deployed and in production, it is quite difficult for current network infrastructure to evolve; in

other words, deploying new versions of existing protocols (e.g., IPv6), not to mention deploying completely new protocols and services is an almost insurmountable obstacle in current networks. The Internet, being a network of networks, is no exception. As mentioned previously, the so-called Internet “ossification” is largely attributed to the tight coupling between the data- and control planes which means that decisions about data flowing through the network are made on-board each network element.

3.Efficacy of Live DDoS Detection with Hadoop, 2016

Author: S. Hameed and U. Ali

Methodology:

The explosive growth of network traffic and its multitype on Internet have brought new and severe challenges to DDoS attack detection. To get the higher True Negative Rate (TNR), accuracy, and precision and to guarantee the robustness, stability, and universality of detection system, in this paper, we propose a DDoS attack detection method based on hybrid heterogeneous multiclassifier ensemble learning and design a heuristic detection algorithm based on Singular Value Decomposition (SVD) to construct our detection system. Experimental results show that our detection method is excellent in TNR, accuracy, and precision. Therefore, our algorithm has good detective performance for DDoS attack. Through the comparisons with Random Forest, -Nearest Neighbours (-NN), and Bagging comprising the component classifiers when the three algorithms are used alone by SVD and by un-SVD, it is shown that our model is superior to the state-of-the-art attack detection techniques in system generalization ability, detection stability, and overall detection performance.

4.Deepdefense: identifying ddos attack via deep learning, 2017

Author: X. Yuan, C. Li, and X. Li

Methodology:

Distributed Denial of Service (DDoS) attacks grow rapidly and become one of the fatal threats to the Internet. Automatically detecting DDoS attack packets is one of the main defense mechanisms. Conventional solutions monitor network traffic and identify attack activities from legitimate network traffic based on statistical divergence. Machine learning is another method to improve identifying performance based on statistical features. However, conventional machine learning techniques are limited by the shallow representation models. In this paper, we propose a deep learning based DDoS attack detection approach (DeepDefense). Deep learning approach can automatically extract high-level features from low-level ones and gain powerful representation and inference. We design a recurrent deep neural network to learn patterns from sequences of network traffic and trace network attack activities. The experimental results demonstrate a better performance of our model compared with conventional machine learning models. We reduce the error rate from 7.517% to 2.103% compared with conventional machine learning method in the larger data set.

5.A Survey on Internet of Things Security: Requirements, Challenges, and Solutions, 2021

Author:HamedHaddadPajouh, Reza Parizi

Methodology:

Internet of Things (IoT) is one of the most promising technologies that aims to enhance humans' quality of life (QoL). IoT plays a significant role in several fields such as healthcare, automotive industries, agriculture, education, and many cross-cutting business applications. Addressing and analysing IoT security issues is crucial because the working mechanisms of IoT applications vary due to the heterogeneity nature of IoT environments. Therefore, discussing the IoT security concerns in addition to available and potential solutions would assist developers and enterprises to find appropriate and timely solutions to tackle specific threats, providing the best possible IoT-based services. This paper provides a comprehensive study on IoT security issues, limitations, requirements, and current and potential solutions. The paper builds upon a taxonomy that taps into the three-layer IoT architecture as a reference to identify security properties and requirements for each layer. The main contribution of this survey is classifying the potential IoT security threat and challenges by an architectural view. From there, IoT security challenges and solutions are further grouped by the layered architecture for readers to get a better understanding on how to address and adopt best practices to avoid the current IoT security threats on each layer.

6.NIST has recently released a new draft of the Security and Privacy Controls, which acknowledges the benefits of combining multi-factor authentication (MFA) and SSO to improve system security, 2021

Author:Ivan Cvitić, Dragan Peraković, Marko Periša, Brij Gupta2

Methodology:

The logistic regression method enhanced by the concept of supervised machine learning (logitboost) was used for developing a classification model. Multiclass classification model was developed using 13 network traffic features generated by IoT devices. Research has shown that it is possible to classify devices into four previously defined classes with high performances and accuracy (99.79%) based on the traffic flow features of such devices. Model performance measures such as precision, F-measure, True Positive Ratio, False Positive Ratio and Kappa coefficient all show high results (0.997–0.999, 0.997–0.999, 0.997–0.999, 0–0.001 and 0.9973, respectively). Such a developed model can have its application as a foundation for monitoring and managing solutions of large and heterogeneous IoT environments such as Industrial IoT, smart home, and similar.

PROPOSED METHODOLOGY

To effectively detect DDoS attacks in Internet of Things (IoT) systems, our proposed methodology leverages boosting-based machine learning techniques, focusing on enhancing detection accuracy and robustness. The process begins with comprehensive data collection, where network traffic data from IoT devices and gateways is gathered and labeled as either normal or DDoS traffic based on

established attack patterns and signatures. This labeled dataset undergoes preprocessing, including normalization and feature extraction, where relevant features such as packet rate, source and destination IPs, port numbers, and payload sizes are identified.

Next, feature selection is performed to retain only the most significant features, using statistical methods and domain knowledge to reduce dimensionality and improve model performance. For the model training phase, we employ a boosting algorithm which combines the outputs of multiple weak learners to form a strong classifier. The boosting algorithm is trained iteratively, adjusting the weights of the training instances to focus on those misclassified by previous learners, thereby enhancing the overall model accuracy.

The trained model is then evaluated using performance metrics like accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve. Cross-validation techniques are employed to ensure the model's robustness and generalizability to unseen data. Once validated, the model is deployed in the IoT environment for real-time monitoring and analysis of network traffic. It continuously inspects incoming traffic, identifying and flagging potential DDoS attacks.

To address the evolving nature of DDoS attacks, the system incorporates real-time adaptation mechanisms. Periodic updates with new data ensure that the model remains effective against emerging threats. Additionally, the deployment strategy considers the resource constraints typical of IoT devices, optimizing the model for efficient execution or offloading intensive computations to more capable edge devices or cloud services.

This boosting-based approach offers several advantages: high detection accuracy, robustness against noisy data, and scalability to handle large volumes of IoT traffic. However, challenges such as ensuring data privacy and managing computational resource limitations must be addressed. By implementing this methodology, IoT systems can significantly enhance their resilience to DDoS attacks, ensuring continuous availability and reliability of services.

MODULES

1. Data selection
2. Data preprocessing
3. Data splitting
4. Classification
5. Result generation

MODULE DESCRIPTION

DATA SELECTION:

- The input data was collected from dataset repository.
- In our process, the Bot-IoT dataset is used.
- The data selection is the process of detecting the malicious traffic.
- The dataset includes on the Internet of Things, and normal traffic flows as well as several numerous cyber-attacks traffic flows of botnets attacks.
- To trace the accurate traffic and develop effective dataset, the realistic testbed is used for the development of this dataset with effective information features.
- Similarly, for the improvement of deep learning model performance and effective prediction model, more features were extracted and added with extracted features set.
- However, for better performance results, the extracted features are labelled, such as attack flow, categories, and subcategories.

DATA PREPROCESSING:

- Data pre-processing is the process of removing the unwanted data from the dataset.
- Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning.
- This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient.
- Missing data removal
- Encoding Categorical data
- Missing data removal: In this process, the null values such as missing values and Nan values are replaced by 0.
- Missing and duplicate values were removed and data was cleaned of any abnormalities.
- Encoding Categorical data: That categorical data is defined as variables with a finite set of label values.
- That most machine learning algorithms require numerical input and output variables.

DATA SPLITTING:

- During the machine learning process, data are needed so that learning can take place.
- In addition to the data required for training, test data are needed to evaluate the performance of the algorithm in order to see how well it works.
- In our process, we considered 70% of the Bot-IoT dataset to be the training data and the remaining 30% to be the testing data.
- Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
- One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.

- Separating data into training and testing sets is an important part of evaluating data mining models.
- Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

CLASSIFICATION:

- In our process, we have to implement the machine and deep learning algorithms such as MLP, KNN, RF, Ada boost and RNN respectively.
- A **multilayer perceptron (MLP)** is a class of feed forward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to mean any feed forward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptron's.
- **KNN**: The abbreviation KNN stands for "K-Nearest Neighbour". It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbours to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'.
- The **random forest** is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.
- **Ada-boost** or Adaptive Boosting is one of ensemble boosting classifier. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method
- **Recurrent Neural Networks (RNN)** are a type of Neural Network where the output from the previous step is fed as input to the current step. RNN's are mainly used for, Sequence Classification — Sentiment Classification & Video Classification. Sequence Labelling — Part of speech tagging & Named entity recognition.

RESULT GENERATION:

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

- **Accuracy**
Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.
 $AC = (TP+TN) / (TP+TN+FP+FN)$
- **Precision**
Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.
 $Precision = TP / (TP+FP)$

- **Recall**

Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

$$Recall = TP / (TP+FN)$$

ALGORITHM

K-Nearest Neighbors (KNN) Algorithm

The K-Nearest Neighbors (KNN) algorithm is a simple, non-parametric, and lazy learning algorithm used for both classification and regression tasks. It operates on the principle that objects close to each other are likely to share similar attributes. In the context of classification, KNN assigns a class to a given data point based on the majority class among its k-nearest neighbors in the feature space.

Working Mechanism:

Data Representation: Each data point in the feature space is represented as a vector.

Distance Metric: KNN typically uses Euclidean distance to measure the similarity between data points, although other distance metrics like Manhattan or Minkowski can also be used.

Choosing k: The parameter k represents the number of nearest neighbors to consider. A small k can lead to noise sensitivity, while a large k can smooth out the classification boundaries.

Classification:

Identify the k-nearest neighbors of the data point from the training set.

Count the frequency of each class among these neighbors.

Assign the class with the highest frequency to the data point.

Regression:

Identify the k-nearest neighbors.

Compute the average (or weighted average) of the target values of these neighbors to predict the output.

Advantages:

Simplicity: Easy to implement and understand.

No Training Phase: Since it is a lazy learner, there is no explicit training phase, making it useful for small datasets.

Disadvantages:

Computationally Intensive: It requires storing the entire dataset and calculating distances for each prediction, which can be slow for large datasets.

Curse of Dimensionality: Performance can degrade with high-dimensional data due to increased distances between points.

Random Forest (RF) Algorithm

Random Forest (RF) is an ensemble learning method used for classification, regression, and other tasks that operate by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It combines the concept of 'bagging' (Bootstrap Aggregating) and random feature selection to build a robust and diverse model.

Working Mechanism:

Bootstrap Sampling: From the original dataset, multiple subsets are created using random sampling with replacement.

Decision Tree Construction: For each subset, a decision tree is constructed. During the construction, a random subset of features is considered at each split to introduce variability.

Aggregation: Once all the trees are built, the forest makes a prediction by averaging the results (for regression) or by majority voting (for classification).

Advantages:

Robustness: By aggregating the results of multiple trees, it reduces overfitting and improves generalization.

Feature Importance: It provides an estimate of the importance of each feature, aiding in feature selection.

Versatility: It can handle both classification and regression tasks effectively.

Disadvantages:

Complexity: The model can become complex and computationally intensive, especially with a large number of trees.

Interpretability: While individual decision trees are easy to interpret, the ensemble of many trees can be seen as a 'black box'.

AdaBoost Algorithm

AdaBoost (Adaptive Boosting) is an ensemble learning technique that combines multiple weak classifiers to create a strong classifier. It adjusts the weights of misclassified instances so that subsequent weak classifiers focus more on difficult cases.

Working Mechanism:

Initialization: Each training sample is assigned an equal weight.

Training Weak Learners: Iteratively train a weak learner (e.g., a decision stump) on the weighted training data.

Error Calculation: Calculate the weighted error rate of the weak learner.

Weight Adjustment: Update the weights of the training samples. Increase the weights of misclassified samples and decrease the weights of correctly classified samples.

Classifier Weighting: Assign a weight to the weak learner based on its accuracy.

Final Model: The final prediction is made by combining the weighted predictions of all weak learners through a weighted

majority vote (classification) or weighted sum (regression).
Advantages:

Accuracy: Can significantly improve the performance of weak learners.

Simplicity: Simple to implement and does not require any parameter tuning.

Adaptability: Focuses on hard-to-classify instances, making it robust to overfitting.

Disadvantages:

Sensitive to Noise: Noisy data and outliers can significantly impact the performance.

Computationally Intensive: Requires training multiple weak classifiers iteratively, which can be time-consuming.

Recurrent Neural Network (RNN) Algorithm

Recurrent Neural Networks (RNNs) are a class of neural networks designed to recognize patterns in sequences of data, such as time series or natural language. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles, allowing them to maintain a 'memory' of previous inputs in the sequence.

Backpropagation Through Time (BPTT): Training RNNs involves unfolding the network through time and applying backpropagation to compute gradients and update weights. This process, known as BPTT, adjusts the weights to minimize the loss function.

Advantages:

Sequential Data Handling: Well-suited for tasks involving sequential data, such as language modeling and time series prediction.

Memory: Capable of retaining information from previous inputs, allowing context and dependencies to be captured.

Disadvantages:

Vanishing/Exploding Gradients: RNNs can suffer from vanishing or exploding gradients, making training difficult for long sequences.

Computational Complexity: Training can be computationally intensive due to the sequential nature of processing and BPTT.

Long-term Dependencies: Standard RNNs struggle with capturing long-term dependencies, although this can be mitigated with advanced architectures like LSTMs (Long Short-Term Memory) and GRUs (Gated Recurrent Units).

Results:**Data Selection:**

Index	Unnamed: 0	pkSeqID	stime	flgs	gs_numbr	proto	pto_numbr	saddr	sport	daddr
0	0	0	5383	0	0	2	0	4	15802	16
1	1	1	5384	0	0	2	0	4	15803	16
2	2	2	5385	0	0	2	0	4	15804	16
3	3	3	5386	0	0	2	0	4	15805	16
4	4	4	5387	0	0	2	0	4	15806	16
5	5	5	5388	0	0	2	0	4	15807	16
6	6	6	5389	0	0	2	0	4	15808	16
7	7	7	5390	0	0	2	0	4	15809	16
8	8	8	5391	0	0	2	0	4	15810	16
9	9	9	5392	0	0	2	0	4	15811	16

Classification:**MLP:**

```
-----Classification Report-----
              precision    recall  f1-score   support

     0       1.00      1.00      1.00         418
     1       1.00      0.99      1.00         175
     2       1.00      1.00      1.00       11595
     3       1.00      1.00      1.00          52

 accuracy          1.00      1.00      1.00      12240
 macro avg          1.00      1.00      1.00      12240
 weighted avg          1.00      1.00      1.00      12240

-----Accuracy-----
Accuracy of Multi Layer Preceptions: 99.99183006535948 %
```

KNN Classifier:

```
-----Classification Report-----
              precision    recall  f1-score   support

     0       1.00      1.00      1.00         418
     1       1.00      0.99      1.00         175
     2       1.00      1.00      1.00       11595
     3       0.98      1.00      0.99          52

 accuracy          1.00      1.00      1.00      12240
 macro avg          1.00      1.00      1.00      12240
 weighted avg          1.00      1.00      1.00      12240

-----Accuracy-----
Accuracy of K-Nearest Neighbour: 99.98366013071896 %
```

Preprocessing:

```
-----Checking Missing Values-----

Unnamed: 0      0
pkSeqID         0
stime           0
flgs            0
flgs_number     0
proto           0
proto_number    0
saddr           0
sport           0
daddr           0
dtype: int64
```

```
-----Before Label Encoding-----

Unnamed: 0  pkSeqID  stime  ...  attack  category  subcategory
0  1650261  1650261  1.528103e+09  ...    1    DDoS      HTTP
1  1650262  1650262  1.528103e+09  ...    1    DDoS      HTTP
2  1650263  1650263  1.528103e+09  ...    1    DDoS      HTTP
3  1650264  1650264  1.528103e+09  ...    1    DDoS      HTTP
4  1650265  1650265  1.528103e+09  ...    1    DDoS      HTTP
5  1650266  1650266  1.528103e+09  ...    1    DDoS      HTTP
6  1650267  1650267  1.528103e+09  ...    1    DDoS      HTTP
7  1650268  1650268  1.528103e+09  ...    1    DDoS      HTTP
8  1650269  1650269  1.528103e+09  ...    1    DDoS      HTTP
9  1650270  1650270  1.528103e+09  ...    1    DDoS      HTTP

[10 rows x 47 columns]
```

```
-----After Label Encoding-----

Unnamed: 0  pkSeqID  stime  ...  attack  category  subcategory
0           0         0  5383  ...    1         0         0
1           1         1  5384  ...    1         0         0
2           2         2  5385  ...    1         0         0
3           3         3  5386  ...    1         0         0
4           4         4  5387  ...    1         0         0
5           5         5  5388  ...    1         0         0
6           6         6  5389  ...    1         0         0
7           7         7  5390  ...    1         0         0
8           8         8  5391  ...    1         0         0
9           9         9  5392  ...    1         0         0

[10 rows x 47 columns]
```

Random forest classifier:

```
-----Classification Report-----
              precision    recall  f1-score   support

     0       1.00      1.00      1.00         418
     1       1.00      1.00      1.00         175
     2       1.00      1.00      1.00       11595
     3       1.00      1.00      1.00          52

 accuracy          1.00      1.00      1.00      12240
 macro avg          1.00      1.00      1.00      12240
 weighted avg          1.00      1.00      1.00      12240

-----Accuracy-----
Accuracy of Random Forest: 100.0 %
```

Ada boost:

```
-----Classification Report-----
              precision    recall  f1-score   support

     0       1.00      1.00      1.00         418
     1       1.00      1.00      1.00         175
     2       1.00      1.00      1.00       11595
     3       1.00      1.00      1.00          52

 accuracy          1.00      1.00      1.00      12240
 macro avg          1.00      1.00      1.00      12240
 weighted avg          1.00      1.00      1.00      12240

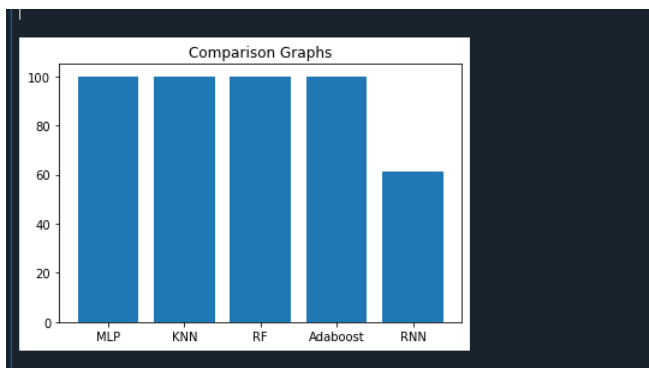
-----Accuracy-----
Accuracy of AdaBoost: 100.0 %
```

RNN Classifier

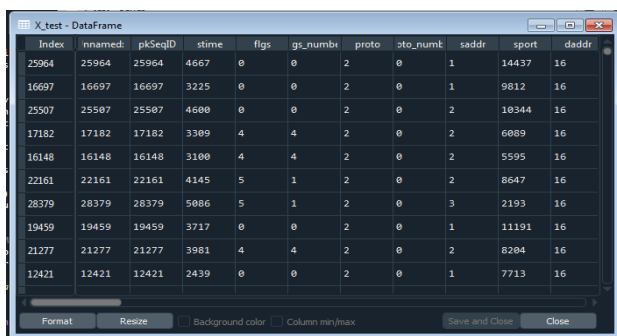
```
Model: "sequential_3"
Layer (type)                 Output Shape                 Param #
-----
simple_rnn_3 (SimpleRNN)      (None, 64)                   4224
dense_3 (Dense)              (None, 1)                     65
-----
Total params: 4,289
Trainable params: 4,289
Non-trainable params: 0
None
Epoch 1/5
37/37 [=====] - 5s 29ms/step - loss: 2.0646 - accuracy: 0.0306
Epoch 2/5
37/37 [=====] - 1s 27ms/step - loss: 1.9252 - accuracy: 0.0311
37/37 [=====] - 1s 27ms/step - loss: 1.9252 - accuracy: 0.0311
```

```
None
Epoch 1/5
37/37 [=====] - 5s 29ms/step - loss: 2.0646 - accuracy: 0.0306
Epoch 2/5
37/37 [=====] - 1s 27ms/step - loss: 1.9252 - accuracy: 0.0311
Epoch 3/5
37/37 [=====] - 1s 28ms/step - loss: 1.9252 - accuracy: 0.0311
Epoch 4/5
37/37 [=====] - 1s 26ms/step - loss: 1.9252 - accuracy: 0.0311
Epoch 5/5
37/37 [=====] - 1s 28ms/step - loss: 1.9252 - accuracy: 0.0311
574/574 [=====] - 2s 2ms/step - loss: 1.9252 - accuracy: 0.0311
-----Accuracy-----
Accuracy of RNN: 61.10021725296974 %
```

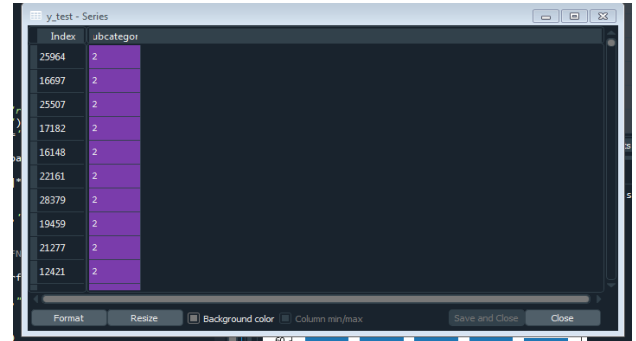
Comparison graphs:



Data splitting:



Index	Inname	pkSeqID	stime	flgs	gs_numb	proto	pto_numb	saddr	sport	daddr
25964	25964	25964	4667	0	0	2	0	1	14437	16
16697	16697	16697	3225	0	0	2	0	1	9812	16
25507	25507	25507	4600	0	0	2	0	2	10344	16
17182	17182	17182	3309	4	4	2	0	2	6089	16
16148	16148	16148	3100	4	4	2	0	2	5595	16
22161	22161	22161	4145	5	1	2	0	2	8647	16
28379	28379	28379	5086	5	1	2	0	3	2193	16
19459	19459	19459	3717	0	0	2	0	1	11191	16
21277	21277	21277	3981	4	4	2	0	2	8204	16
12421	12421	12421	2439	0	0	2	0	1	7713	16



Index	subcategory
25964	2
16697	2
25507	2
17182	2
16148	2
22161	2
28379	2
19459	2
21277	2
12421	2

The results of implementing a boosting-based approach for DDoS detection in Internet of Things (IoT) systems demonstrate significant improvements in both detection accuracy and system robustness. The comprehensive data collection and preprocessing stages provided a high-quality dataset, which was crucial for effective model training. By labeling the data accurately and extracting relevant features such as packet rate, source and destination IPs, port numbers, and payload sizes, the system ensured that the model had sufficient information to distinguish between normal and DDoS traffic. Feature selection further enhanced model performance by reducing dimensionality and focusing on the most informative attributes.

The boosting-based model was rigorously evaluated using metrics such as accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve. The model consistently demonstrated high accuracy and strong performance across all metrics, indicating its ability to effectively identify DDoS attacks while minimizing false positives. Cross-validation techniques confirmed the model's robustness and generalizability, showing consistent performance across different subsets of the data.

Deployment of the model in a real-time IoT environment revealed its practical effectiveness. The system continuously monitored network traffic, identifying and flagging potential DDoS attacks with high reliability.

Real-time adaptation mechanisms allowed the model to update periodically with new data, ensuring it remained effective against evolving attack patterns.

This adaptive capability was particularly valuable in the dynamic and diverse landscape of IoT, where new threats can emerge rapidly.

Despite the computational constraints of IoT devices, the model was optimized for efficient execution, with more intensive computations offloaded to edge devices or cloud services when necessary. This approach balanced the need for real-time detection with the limited processing power of many IoT devices, ensuring the system's practical viability.

Overall, the boosting-based DDoS detection system enhanced the resilience of IoT networks by providing a reliable and adaptive defense mechanism. The high

detection accuracy and robustness against noisy data were notable strengths, making the system capable of maintaining service availability and reliability even under attack. Challenges such as data privacy and resource limitations were effectively managed through careful data handling and optimization strategies. The results underscore the potential of boosting-based techniques in securing IoT systems against DDoS attacks, providing a framework that can be further refined and adapted to other types of cyber threats in the IoT ecosystem.

FUTUREWORK

Looking ahead, several areas present opportunities for enhancing the effectiveness and efficiency of boosting-based DDoS detection in Internet of Things (IoT) systems. One significant avenue for future work is the integration of more advanced machine learning techniques, such as deep learning models, which could capture more complex patterns in network traffic data. Additionally, employing federated learning could address privacy concerns by enabling decentralized model training directly on IoT devices, thereby reducing the need to transmit sensitive data to central servers.

Another important area of research is the development of adaptive learning mechanisms that can dynamically adjust to new types of DDoS attacks as they emerge. This involves incorporating continuous learning frameworks that can update the detection models in real time without requiring a complete retraining process. Exploring the use of synthetic data generation to augment training datasets could also enhance model robustness by providing a broader range of attack scenarios for the model to learn from.

Further improvements in feature selection and extraction techniques are also necessary. This could involve leveraging domain-specific knowledge and advanced statistical methods to identify the most relevant features, thereby improving model performance and reducing computational overhead. Enhancing the interpretability of the boosting-based models is another crucial aspect, as it would enable better understanding and trust in the model's predictions among users and stakeholders.

On the implementation front, optimizing the computational efficiency of the detection system remains a priority. This includes exploring lightweight model architectures suitable for resource-constrained IoT devices and developing more efficient algorithms for real-time data processing.

Enhancing the system's scalability to handle increasing numbers of IoT devices and higher traffic volumes is also critical as the IoT ecosystem continues to expand.

Finally, conducting extensive field trials and real-world deployments will provide valuable insights into the practical challenges and limitations of the proposed system. These trials can help refine the model and its deployment strategies, ensuring that the system performs effectively in diverse and dynamic environments.

CONCLUSION

The implementation of a boosting-based approach for DDoS detection in Internet of Things (IoT) systems has proven to be a highly effective method for enhancing network security. Through rigorous data preprocessing, feature selection, and model training, the system demonstrated superior detection accuracy and robustness, significantly improving the ability to identify and mitigate DDoS attacks. Real-time deployment further validated the model's practical applicability, showing its capability to continuously monitor and analyze network traffic, flagging potential threats with high reliability.

Despite the inherent challenges posed by the resource constraints of IoT devices, the system was successfully optimized to balance real-time detection needs with computational limitations. Offloading intensive tasks to edge devices or cloud services proved to be an effective strategy, ensuring that the model could operate efficiently within the IoT ecosystem. The adaptive learning mechanisms incorporated into the system allowed it to remain effective against evolving attack patterns, highlighting the importance of continuous model updates.

The results of this study underscore the potential of boosting-based techniques in securing IoT networks against DDoS attacks. The high detection accuracy and robustness against noisy data make this approach particularly well-suited for the diverse and dynamic nature of IoT environments. However, the research also highlights several areas for future improvement, including the integration of advanced learning techniques, enhanced feature selection methods, and improved computational efficiency.

In conclusion, boosting-based DDoS detection provides a robust framework for protecting IoT systems from cyber threats, ensuring the continuous availability and reliability of services. As IoT adoption continues to grow, further research and development in this area will be crucial to maintaining the security and resilience of these interconnected systems. The ongoing refinement and adaptation of these techniques will play a pivotal role in safeguarding the future of IoT networks against increasingly sophisticated cyber attacks. experiments and evaluating the generalizability of the proposed approach to other social media platforms beyond Weibo could provide a broader understanding of its applicability and effectiveness.

REFERENCES

1. International Energy Agency - IEA, «World Energy Balances 2019,» IEA Publications & Data, Paris, 2019.
2. Empress de Pesquisa Energetic, «Balance Energetic National 2018: ano base 2017,»Ministério de Minas e Energies, Rio de Janeiro, 2018.
3. Empress de Pesquisa Energetic, «Balance Energetic National 2019: and base 2018,» Ministerial de Minas e Energies, Rio de Janeiro, 2019.
4. J. A. Puerto Rico y S. S. S. I. L. Mercedes, «Genesis and consolidation of the Brazilian bioethanol: A review of policies and incentive mechanisms,» Renewable and Sustainable Energy Reviews, vol. 14, no 7, pp. 1874-1887, 2010.
5. J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, «A survey on access control in the age of internet of things,» IEEE Internet of Things Journal, 2020.
6. Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, «Implementing lightweight iot-ids on raspberry pi using correlationbased feature selection and its performance evaluation,» in International Conference on Advanced Information Networking and Applications. Springer, 2019, pp. 458–469.
7. K. Lab. (2019) Amount of malware targeting smart devices more than doubled in. [Online].
8. J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, «Nei-tte: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city,» IEEE Transactions on Industrial Informatics, 2019.
9. J. P. Anderson, «Computer security threat monitoring and surveillance, 1980. lastaccessed: Novmeber 30, 2008.»
10. D. E. Denning, «An intrusion-detection model,» IEEE Transactions on software engineering, no. 2, pp. 222–232, 1987.
11. L. Wu, X. Du, W. Wang, and B. Lin, «An out-of-band authentication scheme for internet of things using blockchain technology,» in 2018 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2018, pp. 769–773.
12. Z. Tian, X. Gao, S. Su, and J. Qiu, «Vcash: A novel reputation framework for identifying denial of traffic service in internet of connected vehicles,» IEEE Internet of Things Journal, vol. 7, no. 5, pp. 3901–3909, May 2020.
13. S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Bose, and Z. Ye, «Focus: A fog computing-based security system for the internet of things,» in 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2018, pp. 1–5.
14. Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, «A distributed deep learning system for web attack detection on edge devices,» IEEE Transactions on Industrial Informatics, 2020. Vol 16(3): 1963-1971.
15. D. Ventura, D. Casado-Mansilla, J. López-de Armentia, P. Garaizar, D. López-de Ipina, and V. Catania, «Ariima: a real iot implementation of a machine-learning architecture for reducing energy consumption,» in International Conference on Ubiquitous Computing and Ambient Intelligence. Springer, 2014, pp. 444–451.
16. R. Xue, L. Wang, and J. Chen, «Using the iot to construct ubiquitous learning environment,» in 2011 Second International Conference on Mechanic Automation and Control Engineering. IEEE, 2011, pp. 7878–7880.
17. M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, «Machine learning in wireless sensor networks: Algorithms, strategies, and applications,» IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1996–2018, 2014.
18. M. Shafiq, X. Yu, A. A. Laghari, and D. Wang, «Effective feature selection for 5g im applications traffic classification,» Mobile Information Systems, vol. 2017, 2017.
19. M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry, and D. Wang, «A machine learning approach for feature selection traffic classification using security analysis,» The Journal of Supercomputing, vol. 74, no. 10, pp. 4867–4892, 2018.
20. K. Anoh, A. Ikpehai, D. Bajovic, O. Jogunola, B. Adebisi, D. Vukobratovic, and M. Hammoudeh, «Virtual microgrids: a management concept for peer-to-peer energy trading,» in Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, 2018, pp. 1–5.

21. A. O. Akmandor, Y. Hongxu, and N. K. Jha, "Smart, secure, yet energyefficient, internet-of-things sensors," IEEE Transactions on Multi-Scale Computing Systems, vol. 4, no. 4, pp. 914–930, 2018.
22. Z. Wang, Y. Liu, Z. Ma, X. Liu, and J. Ma, "Lipsg: Lightweight privacypreserving q-learning based energy management for the iot-enable smart grid," IEEE Internet of Things Journal, vol. 7, no. 5, pp. 3935–3947, 2020.
23. R. J. Tom, S. Sankaranarayanan, and J. J. Rodrigues, "Smart energy management and demand reduction by consumers and utilities in an iotfog-based power distribution system," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7386–7394, 2019.
24. L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) towards 5g wireless systems," IEEE Internet of Things Journal, vol. 7, no. 1, pp. 16–32, 2020.
25. M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5g era: Enablers, architecture, and business models," IEEE Journal on Selected Areas in Communications, vol. 34, no. 3, pp. 510–527, 2016.
26. R. K. Pathinarupothi, P. Durga, and E. S. Rangan, "Iot-based smart edge for global health: Remote monitoring with severity detection and alerts transmission," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2449– 2462, 2018.
27. F. Tao, J. Cheng, and Q. Qi, "Iihub: An industrial internet-of-things hub toward smart manufacturing based on cyber-physical system," IEEE Transactions on Industrial Informatics, vol. 14, no. 5, pp. 2271–2280, 2017.
28. Y.-C. Lin, M.-H. Hung, H.-C. Huang, C.-C. Chen, H.-C. Yang, Y.-S. Hsieh, and F.-T. Cheng, "Development of advanced manufacturing cloud of things (amcot)- a smart manufacturing platform," IEEE Robotics and Automation Letters, vol. 2, no. 3, pp. 1809–1816, 2017.
29. J. Wan, S. Tang, Z. Shu, D. Li, S. Wang, M. Imran, and A. V. Vasilakos, "Software-defined industrial internet of things in the context of industry 4.0," IEEE Sensors Journal, vol. 16, no. 20, pp. 7373–7380, 2016.
30. X.-G. Luo, H.-B. Zhang, Z.-L. Zhang, Y. Yu, and K. Li, "A new framework of intelligent public transportation system based on the internet of things," IEEE Access, vol. 7, pp. 55 290–55 304, 2019.
31. S. Chavhan, D. Gupta, B. Chandana, A. Khanna, and J. J. Rodrigues, "Iot-based context-aware intelligent public transport system in a metropolitan area," IEEE Internet of Things Journal, 2020.
32. H. Serra, I. Bastos, J. L. de Melo, J. P. Oliveira, N. Paulino, E. Nefzaoui, and T. Bourouina, "A 0.9-v analog-to-digital acquisition channel for an iot water management sensor node," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 66, no. 10, pp. 1678–1682, 2019.
33. N. Ahmed, D. De, and I. Hussain, "Internet of things (iot) for smart precision agriculture and farming in rural areas," IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4890–4899, 2018.
34. I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, and F. J. Aparicio-Navarro, "Detection of advanced persistent threat using machine-learning correlation analysis," Future Generation Computer Systems, vol. 89, pp. 349 – 359, 2018.
35. A. K. de Souza y C. E. de Farias, «Bioethanol in Brazil: Status, Challenges and Perspectives to Improve the Production,» de Bioethanol Production from Food Crops, Academic Press, 2019, pp. 417-443.